Trust Model for Certificate Revocation in Ad hoc Networks

Abstract

In this paper we propose a distributed trust model for certificate revocation in Ad-hoc networks. The proposed model allows trust to be built over time as the number of interactions between nodes increase. Furthermore, trust in a node is defined not only in terms of its potential for maliciousness, but also in terms of the quality of the service it provides. Trust in nodes where there is little or no history of interactions is determined by recommendations from other nodes. If the nodes in the network are selfish, trust is obtained by an exchange of portfolios. Bayesian networks form the underlying basis for this model.

1 Introduction

An Ad hoc wireless network or Manet (Mobile Ad hoc NETwork) is formed dynamically by a group of moving nodes. It has no infrastructure and the network has no backbone or a central point of communication [8]. Such networks find applications in scenarios where it is very difficult to establish base stations and where timely updating and maximum mobility is required. These include conferences, military battlefields, disaster/rescue operations, civil enforcement and home networking. The devices operating in an ad hoc network have limited energy. Hence any protocol or administrative operation must be devised with a view of minimizing the battery usage. The limited battery power may make the nodes in the network selfish, i.e. they may not forward or reply to some messages in order to save battery power.

The two main challenges facing ad hoc wireless networks are quality of service and security. The nodes in the ad hoc network operate over wireless media making them highly vulnerable to attacks and the limited bandwidth results in degraded quality of service. Various techniques have been proposed to ensure security and Quality of Service. For example, in order to ensure security, each node in the network may be authenticated using cryptographic techniques supported by a certificate authority (CA). Rather than propose new security or Quality-of-Service mechanisms, in this paper we investigate the role of trust in ensuring security and good service. All nodes in the network should not be considered to be equally trustworthy from a security or service perspective. To ensure a reliable network, it is essential to identify nodes which are trustworthy and those which are unreliable from a security or a service aspect.

This paper introduces a Bayesian network trust model for ad hoc networks which is based on the localized trust model introduced in [6]. Establishing trust among the nodes in an ad hoc network is very important from a security standpoint as the nodes act as self securing devices to protect themselves without any infrastructural support. Furthermore, nodes route packets to a destination through intermediate nodes. Hence, nodes need assurance to rely on other nodes in the network and this is achieved by establishing trust

relationships among the nodes. The model discussed in this paper establishes trust relationships among the nodes in the network; the trust values are propagated and evolved in the network with the help of the Bayesian model that each node maintains for every other node in the network. Trust in this work is defined both from a quality perspective, that is, the quality of a node as a service provider, and from a security perspective, that is, the maliciousness of a node. A good trust model should be scalable; it should have an extensive adversary control mechanism and should establish trust among the nodes. However, establishing trust relationships among the nodes involves communication overheads. The primary objective of this work is that trustworthy nodes, that is, nodes that provide good quality of service and are reliable from a security perspective, should survive in the network, whereas nodes which do not provide a good quality of service or are malicious should be detected quickly and removed from the network. This paper shows that these trust relationships help in removing malicious nodes from the network and that establishing trust relationships among the nodes in the network results in better performance and security in the network. The main contribution of this paper is the development of a trust model for ad hoc networks that takes into account attacks and intrusions, the availability of other nodes in the network as well as the quality of service provided other nodes. Section 2 discusses the related work in this area, in section 3, we present the proposed approach, section 4 evaluates the model with simulations and section 5 concludes the paper.

2 Related Work

A number of trust models for ad hoc networks have been proposed. These include a number of approaches to establishing trust in routing protocols. In [9] a trust based adaptive on demand ad hoc routing protocol is proposed where based upon the trust that a node has on its neighboring node, a security level is implemented. In this work the security level (or level of encryption) is defined by the level of trust between nodes. However, how the trust between nodes is determined is not discussed. A trust evaluation based security solution [16] is proposed to provide effective security decision for secure routing. Each node's evaluation of trust on other nodes is based on trust factors as experience statistics, data value, intrusion detection result, as well as node owner's preference and policy. In this model only direct experiences are considered and the experiences of other nodes is not taken into account. [17] dynamically updates trust levels by using reports from Intrusion Detection Systems (IDSs) located on all nodes in the network. The nodes neighboring to a node exhibiting suspicious behavior initiate trust reports. Using these trust levels as a guide, the source node then selects a route that meets the security requirements of the message to be transmitted. The trust is determined solely based on intrusions detected. Pirzada et al [10] propose a trust model based on analyzing network packet data. In particular, information such as passive acknowledgments, packet precision, blacklists, salvaging information and gratuitous route replies are examined. The DSR routing protocol [7] is extended to finding trustworthy routes. These protocols either increase the security or base trust on a factor such as intrusions detected, packet data etc. Our objective is to provide a dynamic trust evolution framework that is multi-dimensional, that is, the trust evolves depending on a number of factors.

A number of models have also been proposed where the primary objective is not secure routing. In the model proposed by Cr´epeau and Davis [4], the nodes in the ad hoc

network maintain profile tables. These tables can become large resulting in time consuming exchange of large profile tables. This work focuses on key management for certificate revocation rather than trust evolution. Trust is based on accusations received. In the model proposed by Capra [3] each node exchanges recommendation letters which describe the performance and satisfaction from the node in a specific context. We extend the recommendation mechanism proposed in this paper to include a probabilistic trust approach that is based on a number of factors including maliciousness of a node. Furthermore, the performance or the overheads associated with the approach is not discussed. Buchegger and Le Boudec [1] use a Bayesian network model to detect malicious nodes in a mobile Ad-hoc network on the basis of rumors spread by other nodes. They presented a mechanism to detect potential lies spread by other nodes. This work focuses on building trust based on rumors. Theodorakopoulos and Baras [13] establishes a trust relationship among the nodes based on a theory of semirings. This paper presents a theoretical framework for establishing indirect trust relationships between two nodes without direct interaction. F. Ren et. al. [11] propose a probabilistic solution based on distributed trust model. A secret dealer is introduced in the system bootstrapping phase to complement the assumption in trust initialization. A robust trust chain in then constructed with high probability. This approach depends on a trustworthy secret dealer. The trust model described by Luo [6] is distributed in nature. The Certification authority is based on threshold cryptography [12]. In this model, each node in the network maintains a certificate revocation list, which contains a list of misbehaving nodes and its accusers. If the number of accusers are at least k (the threshold) then the node is marked as convicted otherwise it is marked as a suspect and the certificate is not renewed for a convicted node.

Trust has also been investigated outside of ad hoc mobile networks. Zhu et. al. [18] present a proxy-based approach that uses alternative network channels to establish a secure trust relationship between communication parties to facilitate wireless communications between clients and services. A peer to peer trust model based on satisfaction of interactions is given in [14]. In this model there is no reference to the influence of malicious nodes on trust. Kagal et. al. have proposed an architecture for trust-based security in pervasive computing environments [19]. The architecture is agent-based and built on a role-based framework.

To summarize, a lot of work has been done in trustworthy communications. Trust is typically determined from a security standpoint based on intrusions detected, direct experiences, recommendation from other nodes, accusations etc.. Theoretical models have been proposed and determining trust by analyzing data at the packet level have also been investigated. There is a need for a trust model that will evaluate trust on multiple criteria identified above. Furthermore the trust must evolve based on direct experiences, recommendations and other factors. Although all the above works address some aspect of trust, the work reported here provides an integrated framework for trust that benefits ‘good’ nodes, but handicaps ‘bad’ nodes in a mobile ad hoc network.

3 Trust Model

The proposed trust model for certificate revocation described in this paper meets the following requirements:

- A distributed trust framework
- Trust is built over time as the number of interactions between nodes increase

- Trust in a node is defined not only in terms of its potential for maliciousness, but also in terms of the availability and quality of service it provides
- Trust in nodes where there is little or no history of interactions is determined by recommendations from other nodes
- If the nodes in the network are selfish, trust is obtained by an exchange of portfolios
- The overhead in communications cost by the dissemination of trust information in the network is kept to a minimum.

In a mobile ad hoc network, a source node communicates with a destination node through intermediate nodes. Nodes therefore need to establish trust only with the neighboring nodes with which they communicate directly. Hence trust is established along the path of communication.

In our proposed model, we assume that a certificate authority issues private and public keys to every node that enters the network. In other words, it is assumed that every node is initially trustworthy. A major objective of this work is to show that in the proposed model, nodes which although initially trustworthy, but have turned malicious will be detected and ejected from the network. Detecting such 'insider' malicious nodes is a major challenge for any security system.

3.1 Model of trust

In our proposed work, trust is within the range [0…1] where 1 indicates maximum trust and 0 indicates no trust at all. Trust is a measure of two factors:

- Satisfaction: A node that provides good quality of service (where quality may be defined in terms of response-delay, jitter for example) and is highly available (that is, the neighboring node is available for a long time, such as when the two nodes are close to each other and move in tandem) is more trustworthy than a node that provides poor quality and low availability.
- Maliciousness: A node that is detected to be malicious is deemed to be highly untrustworthy. Some types of malicious activities are unacceptable and the trust in the target node is completely removed and set to 0, whereas other types of malicious activities may be more tolerable and the trust level is reduced, but not completely removed. A node which may be detected spoofing, for example, is defined to be completely untrustworthy and will be set to a trust of 0 (no trust at all).

3.2 Trust Establishment

Trust may be established in three ways in the ad hoc network:

- *Direct experience:* A node can determine trust on a target node based on its previous experience with the target node. This previous experience is only valid for a certain period of time.
- *Seeking recommendation:* If the node doesn't have direct previous experience with the target node or if the experience is outdated the node seeks recommendations. The trust is formulated on the basis of the recommendations received from other nodes.
- *Inadequate trust formation:* Devices that operate in an Ad hoc network are energy constrained. When a request is sent by a source node for recommendations about a particular node, a node may be selfish and not reply so as to conserve its energy. Due to lack of recommendations, the source node is thus unable to determine a reliable trust

level on a particular target node in the network even though the node may be a trustworthy node. Therefore, to deal with such a situation, an exchange of portfolios takes place. In other words, at the end or break of any communication between two nodes, they exchange a "credential" letter based on their experience with each other. Hence, when a node *n* has finished communication with a node *x,* the following exchange takes place:

node *n* to node *x*:

<<"*n*" trusts "*x*", context c_1, time t_s> ,…,<" *n*" trusts "*x*", context c_i, time t_s>, $K\text{-}_n$>.

This means that node *n* trusts node *x* in the context c_i at a time stamp t_s. Context here can be either Quality of Service (QoS) or Availability (A). This credential letter is signed by the private key ($K\text{-}_n$) of the node *n*, thus ensuring that a node will not send fake credentials. Similarly:

node *x* to node *n*:

<<"*x*" trusts "*n*", context c_1, time t_s> ,…,<" *x*" trusts "*n*", context c_i, time t_s>, $K\text{-}_x$>.

Hence when a node encounters a situation where very few nodes reply to a request for recommendations, it directly requests the target node for the portfolio of credentials it has received from other nodes. The target node collects the credentials it has received and creates a "portfolio of credentials". For example:

<<*a* 0.5 *y*, *A*, 100>, <*a* 0.75 *y*, *QoS*, 100>,<*c* 0.25 *y*, *A*, 150>, <*c* 0.40 *y*, *QoS*, 100> $K\text{-}_x$>.

Here node *a* trusts node y for half the communications to be trustworthy as far as availability is concerned and three-quarters for QoS. Node *c*'s trust on node *y* is less. The source node makes a decision on trusting the target node based on the portfolios it receives from the target node.

However a target node may send only selective portfolios to the requesting node. This is very difficult to detect. A partial solution is for the source node to check its local table for inconsistencies. For example, target node *a* sends node *b* its credentials received from nodes *m* and *n*. However *b* knows from prior portfolios it received from *s* that *s* has received a portfolio from *a*. Hence *b* knows that *a* has not sent it the portfolio it has received from s.

An ad hoc network may be set up 'on the fly' and this calls for a key management scheme that is distributed and does not require a centralized key management authority. Although there are a number of such schemes available, each of these schemes require some common security parameter. For example, Zhang et. al. [22] propose a compromise-tolerant security mechanism based on identity-based cryptography and bilinear pairing technology. This is a public/key key management scheme that does not require a centralized public key infrastructure and the public and private keys are generated by the individual nodes. Although this does not compromise the security of the network, it is assumed that the nodes are installed with a common master key at the beginning which can be deleted by the individual nodes once they have generated their public/private keys.

In figure 1, arc 1, refers to the direct experience a node has with another node. This direct experience is based on the availability and quality of service. The satisfaction is derived from the availability and quality of service and then the trust is derived. A node is trustworthy if the satisfaction is above some threshold, otherwise it is not trustworthy. In arc 2, a node derives its trust on another node based on recommendations it receives from other nodes. In arc 3, a node determines the trust level based on the portfolio of credentials it receives from the target node. In arcs 4 and 5 an intrusion is detected. In the case of a

serious intrusion (arc 5), the trust is immediately set to 0, whereas in the case of a minor intrusion (arc 4) the trust level is reduced.

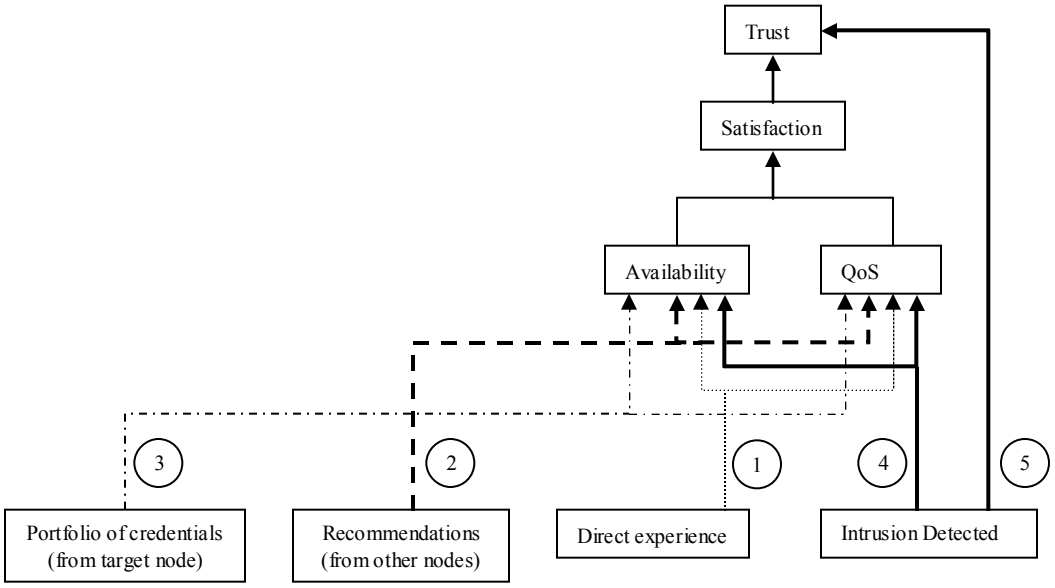

Figure 1: Trust Model

3.3 Intrusion detection unit

Although quality of service and availability are a basis for trust, they do not take into account the maliciousness of nodes. We therefore assume that each node has an intrusion detection unit (IDU) that signals when suspicious activity is detected. It is possible that the behavior of the suspect node may be perfectly legal, and it appears to be malicious due to interference or other problems. The aim of the proposed trust model is that "no innocent node should suffer and no malicious node should survive". If the suspect node's behavior is obvious for it to be marked as malicious, the source node floods the network with accusation messages against it. Nodes in the network which receive the accusation message completely remove their trust in the accused node if the message is flooded by a trustworthy node. On the other hand, if a node is only suspected to be bad, a neighboring node will only reduce its trust on the target node and will not flood an accusation. The objective of this paper is not to describe the IDU or how it works, hence we only outline the main types of attacks modeled in our simulations.

DoS attack (Resource depletion): A simple resource depletion attack model very similar to the one described in [20] is followed where if the requests from a single node are identical and the requests exceed a threshold, an attack is signaled. The IDS detects this scenario by maintaining a count of the number of packets that it receives from each node within a specific time window. If this count crosses a threshold, then an alert is signaled. For simulations, only data packets were transmitted. The IDS can be easily extended to cater for control packets. Figure 2 shows the state transition diagram for a resource depletion attack. On sniffing the first packet between the source and the destination, the node makes a transition from the initial state (INIT) to an intermediate state (IMT). In this state, both a counter and a timer are initialized. The counter is incremented for every packet the node observes between the same source and destination. If the count crosses a threshold within a

specific time interval, the node moves to ATC state and raises an alert. The trust on the misbehaving node is set to zero and an accusation is flooded in the network (ATC state). The counter is reset if the node observes an idle period for a substantial amount of time.

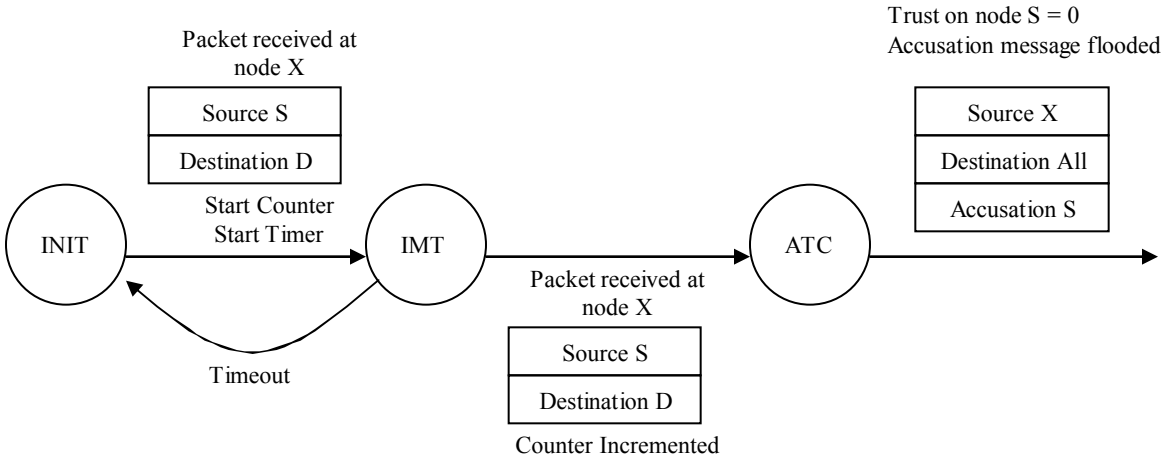

Figure 2: DoS Model

Time of response: This is a quality of service parameter caused by various factors such as congestion etc and is not a security attack due to malicious nodes. There is a maximum waiting time for a response when two nodes are communicating. A timer set to a timeout time T is initiated. The timer decrements and if no response is received with the timeout period T, the IDU will signal and the trust on the node is decreased. No accusation is flooded.

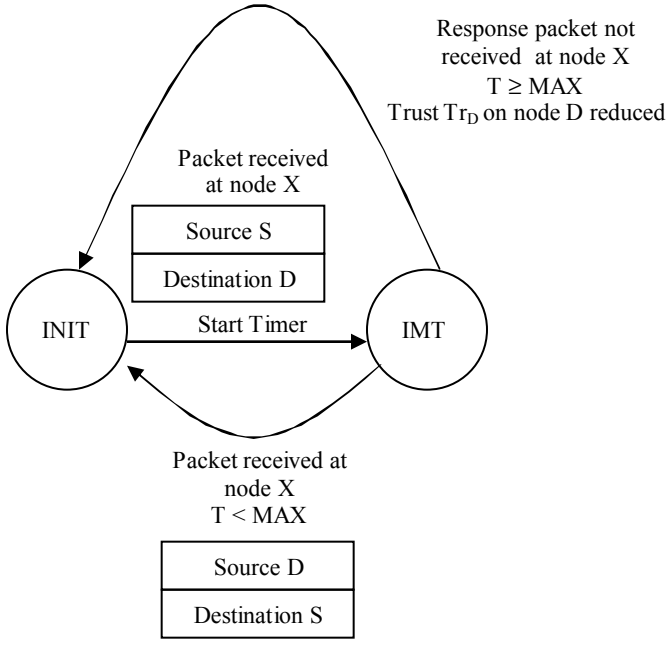

Figure 3: Timer Response model

Spoofing: As mentioned above, nodes exchange credentials (state EXC), which are signed by their private keys. If a node has been spoofing, the credentials cannot be read with the

public key. When the IDU detects a spoofed address, it signals. The trust on the node is completely removed and an accusation message is flooded in the network. This is the approach taken in our model, but this will not detect spoofing in individual packets. Various methods have been proposed for detecting spoofing attacks by analyzing each and every packet [20] [21].

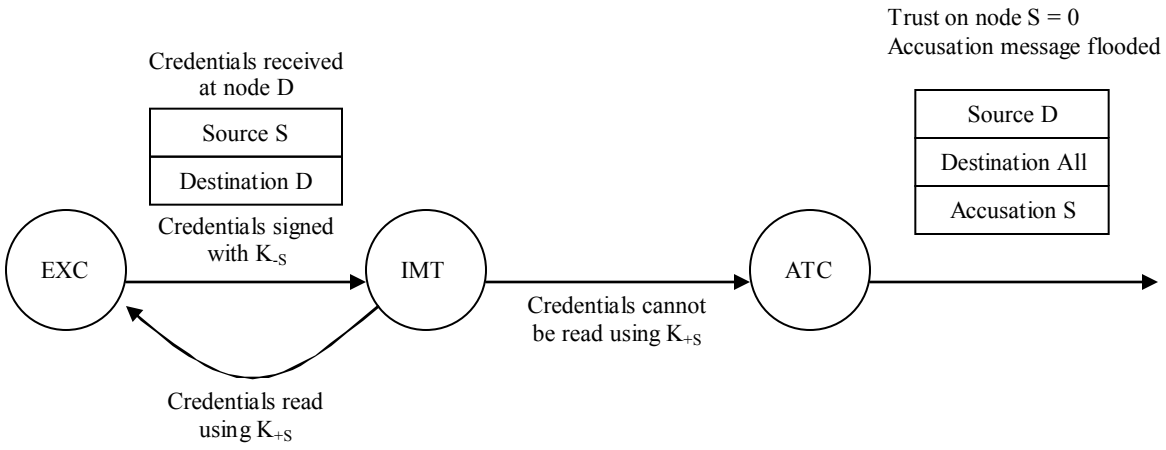

Figure 4: Spoofing model

Hijacking attacks: A malicious node may pick on a victim to falsely accuse in order to remove it from the network. It achieves this by first compromising a node with which it communicates. The compromised node and the malicious node then both accuse an innocent node. The hijacking node repeats this until sufficient number of nodes accuse the victim so that it is removed from the network. A hijacking attack is difficult to detect. For our simulations we modeled a simple detector which raises a flag if the same subset of nodes make the same accusations repeatedly.

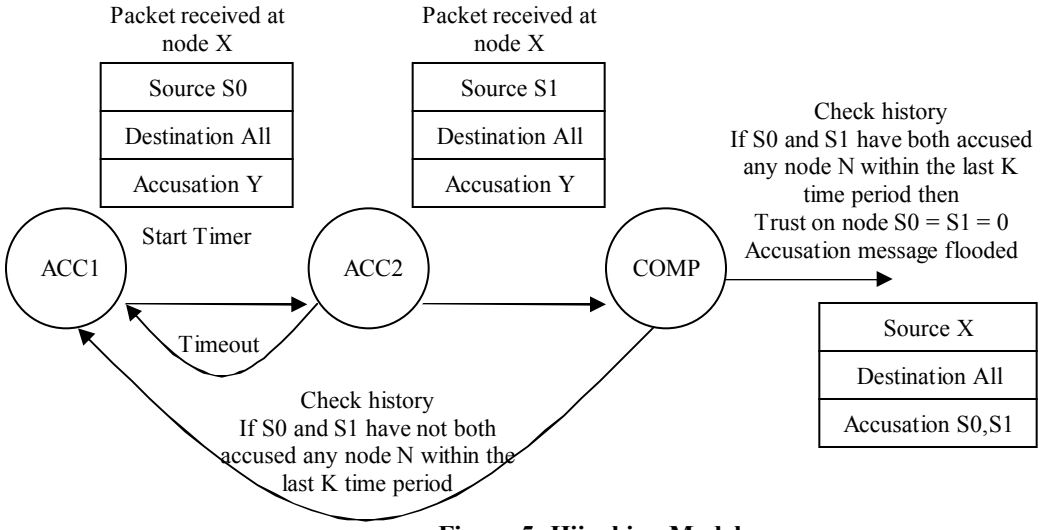

Figure 5: Hijacking Model

An accusation is received in state ACC1. A timer is started. If accusations by multiple nodes are received against the same node within the timeout period, then a comparison is made with a list that contains the nodes that made accusations against a node on previous occasions. This simple correlation is used to determine if nodes are acting in tandem to

make accusations and if such an event is determined, a flag is raised and the node floods the network with accusation messages. In our simple detector, an accusation by the 2 same nodes repeatedly was used to raise a hijacking flag.

3.4 Trust Model

In this work we use a Bayesian network to determine trust in the network. Each node maintains a Bayesian network for every other node in the network. A node in the network will communicate if it has trust on the target neighbor node. A Bayesian network is a relationship network that represents probability relationships between different nodes. A Bayesian network allows trust to be represented using probability which is updated whenever an interaction occurs between two nodes. A node may provide good quality of service, but poor availability. A probability representation that combines these two (or more) contexts of trust deals with such conflicts. The naive Bayes theorem is:

$$P(h \mid e) = \frac{P(e \mid h) * P(h)}{P(e)} \quad (1)$$

P (h) is called the prior probability of hypothesis and $p(e)$ is the prior probability of evidence e. ($h|e$) is the probability of h given e; $P(e|h)$ is the probability of e given h. Hence when all the values are supplied (i.e. prior probabilities) the Bayes theorem computes the posterior probability.

3.4.1 Direct experience

Each node maintains a table of the history of interactions with a target node. This is the history of the number of direct interactions with a particular node, as well as the history of the recommendations received from the node..

Table 1: A table in the node's local environment

Node_id	Direct Interactions					Recommender			
	l_{Total}	Ia	Iq	ls	Time stamp t_s	R_{Total}	Ra	Rq	Rr
A	3	1	1	1	10				
B	2	0	1	1	12				
C	5	5	5	5	15	2	1	1	1

l_a - No of direct experience interactions in which the availability was satisfactory
l_q - No of direct experience interactions in which the QoS was satisfactory
l_s - No. of direct experience interactions that was satisfactory
l_{Total} - Total No. of direct experience interactions
R_a – No. of correct recommendations for Availability
R_q – No. of correct recommendations for QoS
R_r - No. of correct recommendations
R_{Total} – Total No. of recommendations
Each node defines its threshold for satisfaction. For example, during a communication if a neighboring node is available for a time that is greater than a threshold, the availability of that node is defined as satisfactory. However, for the same communication the QoS is below the

threshold for satisfaction. The corresponding entries l_a and l_{Total} in the above table is incremented whereas l_q is not incremented.

The performance of a particular node in the network is stored in the local environment (table 1). Conditional probability is used to find the satisfaction from availability. This is represented as follows:

$$P(Attribute = "Availability" | T = 1) = \frac{P(Attribute = "Avalability", T = 1)}{P(T = 1)} \quad (2)$$

This measures the probability that the attribute is availability given that the interaction is satisfying. Similarly for QoS one can determine that the attribute is QoS given that the interaction is satisfying.

The probability that the interaction is satisfactory is defined as follows:

$$P(T = 1) = \frac{\text{no of satisfying interactions}}{\text{total no of interactions}} = \frac{l_s}{l_{Total}} \quad (3)$$

$P(Attribute="Availability", T=1)$ is the no. of interactions in which the attribute availability was satisfactory for all the interactions that were satisfactory, that is,

$$P(Attribute = "Availability", T = 1) = \frac{l_a}{l_s} \quad (4)$$

The probability that the node is trustworthy in providing Availability can be obtained using the naïve Bayes theorem (eq. (1)):

$$P(T = 1 | Attribute = "Availability") = \frac{P(Attribute = "Availability | T = 1) * P(T = 1)}{P(Attribure = "Availability")} = S_A \quad (5)$$

In eq (5) the probability that the attribute is Availability given that the interaction is satisfying is obtained from eq.(2). The probability that the interaction is satisfactory is determined by eq (3). The probability that Availability is satisfactory is obtained from:

$$P(Attribute = "Availability") = \frac{\text{no of interactions satisfying Availability}}{\text{total no of interactions}} = \frac{l_a}{l_{Total}} \quad (6)$$

Eq (5) shows the probability that the node is trustworthy in providing Availability. Similarly the probability that the node is trustworthy in providing QoS can be derived. The Bayesian model hence serves to build a trust level. After each interaction, a node updates its corresponding Bayesian networks. The parameters l_a, l_q, l_s are incremented appropriately in table 1 and l_{Total} is also incremented. These updated parameters are input to the Bayesian model when determining trust the next time.

Each node has it own perception of trust; hence each node is free to give its own weight for the two parameters. The weight could be any value between [0,1]. A node updates its corresponding Bayesian networks after each interaction. Satisfaction for availability S_A is derived from eq. 5 and similarly satisfaction for quality of service S_Q can be derived. If the satisfaction is above a certain threshold, the node is deemed to be trustworthy and communication takes place. The overall degree of satisfaction with an interaction is computed as:

$$satisfaction(S) = \frac{(W_Q * S_Q + W_A * S_A)}{t_c - t_s} + n \quad (7)$$

where: W_Q: weight for Quality of service

S_Q: Satisfaction from the Quality of service

W_A: weight for availability

S_A: Satisfaction from availability

t_s: value of timestamp of most recent experience
t_c: current time value
n: a constant

The weights W_A and W_Q indicate the relative importance of availability and QoS. Satisfaction S_A and S_Q is a numerical measure of the satisfaction with the availability and the quality of service respectively. If satisfaction S is greater than or equal to a threshold then the node is trustworthy. This node may be selected for communication. All this information is stored in the local environment of the node. The value of timestamp indicates how recent the last experience of the node with the target node was. The more recent direct experiences are given higher weights. A constant n is chosen such that if the satisfaction level is good and then the probability of the node being classified as trustworthy is high. This probability decreases as the most recent direct experience becomes older.

3.4.2 Recommendations

When a node receives from a source node a request for a recommendation on a target node, the node checks its Bayesian network and sends the recommendations for both Availability and Quality of Service. The source node then does two things:

- It first computes a trust level for the target node based on the recommendations it receives.
- At the end of the communication with a target node, it updates the trust level on the recommender, that is, can the recommender be trusted for a recommendation? Each node defines its satisfaction threshold for recommendations. For example, suppose node x receives a positive recommendation about a node y from a node z. After nodes x and y have completed communicating, and the interaction is above the threshold for satisfaction (that is, the interaction is satisfactory), the trust in node z as a recommender is increased. Conversely, if the interaction was not satisfactory the trust in node z as a recommender is decreased.

Each node therefore has two trust measures on a node x:

- trust as a service provider which is determined by the source node's direct experience with the target node
- trust as a recommender which is determined by comparing the direct experience of a node with a target node and the recommendation received from node x about the target node.

3.4.2.1 Computing trust level based on recommendations

The node receives recommendation values on QoS and Availability from other nodes in the network. Some of the recommendations are from nodes with which the source node has had prior direct experience ('prior' nodes) or received previous recommendations ('recommender' nodes), whereas other recommendations are from nodes with which the source node has had no prior experience or has not received any recommendations ('unknown' nodes).

If g recommendations are received from 'unknown' nodes, the recommendation value is calculated as:

$$w_u^{QoS} * \frac{\sum_{j=1}^{g} tu_j^{QoS}}{g} + w_u^{A} * \frac{\sum_{j=1}^{g} tu_j^{A}}{g} \quad (8)$$

where: tu_j^{QoS} = The trust that z^{th} node ('unknown' node) has on j^{th} node (target node) for QoS

tu_j^{A} - The trust that z^{th} node ('unknown' node) has on j^{th} node (target node) for Availability

w_u^{QoS} = weight given to references from 'unknown' nodes for QoS

w_u^{A} = weight given to references from 'unknown' nodes for Availability

If k recommendations are received from nodes with which the source node has had prior direct experience ('prior' nodes), the recommendation value is:

$$w_p^{QoS} * \frac{\sum_{l=1}^{k} tp_l * t_{lj}^{QoS}}{\sum_{l=1}^{k} tp_l^{QoS}} + w_p^{A} * \frac{\sum_{l=1}^{k} tp_l^{A} * t_{lj}^{A}}{\sum_{l=1}^{k} tp_l^{A}} \quad (9)$$

where: tp_l^{QoS} - the trust that the source node has on l^{th} recommender ('prior' node) for QoS based on previous direct experience

tp_l^{A} - the trust that the source node has on l^{th} recommender ('prior' node) for Availability based on previous direct experience

t_{lj}^{QoS} - the trust that l^{th} node ('prior' node) has on j^{th} node (target node) for QoS

t_{lj}^{A} - the trust that l^{th} node ('prior' node) has on j^{th} node (target node) for Availability

w_p^{QoS} = weight given to references from 'prior' nodes for QoS

w_p^{A} = weight given to references from 'prior' nodes for Availability

The trust values are obtained from the trust tables in the respective nodes.

If m recommendations are received from nodes from which the source node has received previous recommendations ('recommender' nodes), the recommendation value is:

$$w_r^{QoS} * \frac{\sum_{l=1}^{m} tr_l^{QoS} * t_{lj}^{QoS}}{\sum_{l=1}^{m} tr_l^{QoS}} + w_r^{A} * \frac{\sum_{l=1}^{m} tr_l^{A} * t_{lj}^{A}}{\sum_{l=1}^{m} tr_l^{A}} \quad (10)$$

where: tr_l^{QoS} - the trust that the source node has on l^{th} recommender ('recommender node') for QoS based on previous recommendations from the l^{th} node

tr_l^{A} - the trust that the source node has on l^{th} recommender ('recommender' node) for Availability based on recommendations from the l^{th} node

t_{lj}^{QoS} - the trust that l^{th} node ('recommender' node) has on j^{th} node (target node) for QoS

t_{lj}^{A} - the trust that l^{th} node ('recommender' node) has on j^{th} node (target node) for Availability

w_r^{QoS} = weight given to references from 'recommender' nodes for QoS

w_r^{A} = weight given to references from 'recommender' nodes for Availability

The trust values are obtained from the trust tables in the respective nodes.

More weight is given to direct experience or references from nodes which have supplied previous recommendations than to unknown references

$$w_p^{QoS} + w_p^{A} + w_u^{QoS} + w_u^{A} + w_r^{QoS} + w_r^{A} = 1$$
$$w_p^{QoS} > w_u^{QoS}\ ;\ w_p^{A} > w_u^{A},\ w_r^{QoS} > w_u^{QoS}\ ;\ w_r^{A} > w_u^{A}$$

The total recommendation value is the sum of all the different recommendations and if the recommendation value is greater than or equal to a pre-defined threshold, communication takes place.

3.4.2.2 Model for updating recommendations

Each node also maintains a Bayesian network for each node to evolve trust on a recommender node. If a node's recommendation is proven to be valid, that is, it is correct, its trust as a recommender is increased. When a node *z* replies to a requesting source node *x* its recommendations on a particular node *y* in the network, *z* sends the requesting node *x* uses the direct experience information about *y* that is stored about that node in its local environment.

To determine that the recommender is node *i* given that the recommendation is correct for availability, that is,

$$P(Attribute = "Availability_{rec\_i}" \mid T = 1) = \frac{P(Attribute = "Availability_{rec\_i}", T = 1)}{P(T = 1)} \quad (11)$$

The probability that the recommendation is correct is defined as follows:

$$P(T = 1) = \frac{\text{no of correct recommendations}}{\text{total no of recommendations}} = \frac{R_r}{R_{Total}} \quad (12)$$

The probability that the recommender *i* is trustworthy in providing recommendations for availability can be obtained using the naïve Bayes theorem (eq. (1)):

$$P(T = 1 \mid Attribute = "Availability_{rec\_i}") = \frac{P(Attribute = "Availability_{rec\_i}" \mid T = 1) * P(T = 1)}{P(Attribute = "Availability_{rec\_i}")} \quad (13)$$

eq (13) shows the probability that a node *i* is trustworthy in recommending availability.

The probability that recommendation is correct is obtained from:

$$P(Attribute = "Availability_{rec\_i}") = \frac{\text{no of recommendations correct for Availability}}{\text{total no of recommendations}} = \frac{R_a}{R_{Total}} \quad (14)$$

Similarly the probability that a node *i* is trustworthy in recommending QoS can be determined.

If a node's recommendation is proven to be valid, its trust as a recommender is increased. The number of correct recommendations R_r, the total number of recommendations R_{Total} and the number of correct recommendations for Availability (R_a) and QoS (R_q) is incremented. Hence each node maintains a trust record of recommenders and service providers (direct experience nodes)

3.4.3 Accusation messages

If a node *x* receives an accusation message accusing node *y* from a node *n*, node *x* either ignores the accusation message or completely removes its trust on the accused node, depending on the trust of the accuser *n* as a service provider (that is, the trust built as a result of direct interactions). If node *x* has a low trust (below a threshold) value of the accuser *n* as a result of previous direct interactions (service provider) with node *n*, the accusation is simply ignored. On the other hand, if the accuser *n* has a high trust (above

the threshold) value, the trust value on the accused node *y* is immediately set to zero, that is, no trust at all. This simple model can be further refined such that the degree of trust degradation in the accused is proportional to the trust level in the accuser.

3.4.4 Certificate Revocation

In the Ad hoc network, each node communicates only for the period the certificate is valid. Once the certificate expires, it has to be renewed. The node broadcasts a request for certificate renewal. Based on threshold cryptography, it should get reply from at least k (threshold) nodes to renew its certificate. When a certificate is to be renewed the node that receives the request, replies with a request to the source node for a "Portfolio of credentials". This defines the overall behavior of the node in the network during the previous certificate validity period. Suppose if *a*'s certificate needs to be renewed, it sends its request to all the other nodes. Then the node *j* on receiving *a*'s request for certificate renewal, requests *a* for its portfolio of credentials, that is, all the credentials it has received. It adds up all the trust values, if the trust is greater than or equal to threshold, then the partial certificate is given. The same procedure is repeated by each node that receives a request from *a*. If *a* is able to get k partial certificates it can form a valid certificate using threshold cryptography for communicating in the network [6]. Sometimes the node may have one or no recommendation, as it has participated in very few (if any) communications. In such a situation, the local environment of the node (node issuing the certificate) is checked to verify if there is any information about any misbehavior of the source or requesting node. If the node has misbehaved and is marked as convicted it is denied a certificate otherwise it is granted a certificate.

3.5 Exchange of Credentials

There are critical points at which credentials should be exchanged. In the first approach nodes should exchange credentials only after the session has ended. This is the approach taken in our simulations. This approach relies on the lower level communication protocols to compensate for packets that have been lost due to route breakage caused by mobility. No credentials are exchanged when there is a break in communications caused by mobility. The danger with this approach is that when there is a break in a link, one of the nodes in the communications may turn malicious. This may not be easily detected, especially if the majority of the communication took place when the node was not malicious. However, since the routing protocol is responsible for constructing a new path and this can typically be done quickly, the probability of a node becoming malicious is minimal. An alternative approach would try to exchange credentials when a route breakage is predicted. This approach will result in a more accurate measure of trust. Since credentials are exchanged before link breaks, this approach will detect malicious nodes faster, but incurs the penalty of increased overheads. In this approach, credentials are exchanged when there is a high probability of a break in the path. One possible solution to recognizing that a path may be about to break is the signal level, that is, to exchange credentials when the signal is below a threshold level. However, although the signal between any two nodes in the path may be weak, they may be moving in parallel. This will result in a continuous exchange of credentials resulting in congestion and minimal message communications.

We outline a possible simple approach to detecting a break in the link. However more work is needed in this area. We define a "exchange region" which has three zones

called the prediction zone, adaptation zone and exchange zone (figure 6). The range of each zone is determined by the strength of the signal. The innermost zone is the prediction zone and the outermost zone is the exchange zone. The boundary between the adaptation zone and the exchange zone is dynamic, whereas all the other boundaries are fixed. The relative velocity of the nodes is determined by measuring the time elapsed for a node to travel relatively from the start of the prediction zone to the end of the prediction zone. If the velocity is maximum, an exchange of credentials takes place. Otherwise, an adaptation zone is defined, that is, a new signal strength defining the range of the adaptation zone is calculated. When the signal strength reaches the end of the adaptation zone an exchange of credentials takes place. Hence, a continuous exchange of credentials does not take place using this approach even if two nodes are moving in parallel. The different zones can be determined based on the received signal strength. The signal power of received messages can be used to estimate the distance between them. In [5] the signal power is determined to be

$$P_{received} = \frac{P_0}{r^4} \tag{15}$$

where r is the distance from the transmitter and P_0 is the transmitted power which is a constant based on antenna gain and height. Various approaches for dealing with link and path breaks have been proposed, [23] for example.

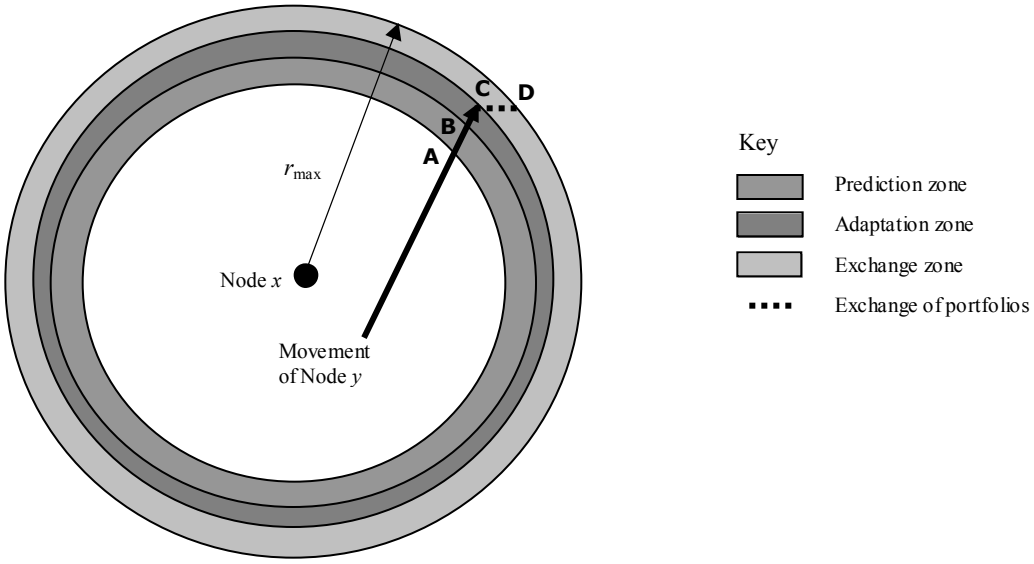

Figure 6: Predicting link breakage

4 Simulations and Results

We compare the proposed Bayesian Trust model with the distributed trust model described in [6]. In this model the distributed certification authority is based on threshold cryptography. Each node in the network maintains a certificate revocation list, which contains a list of misbehaving nodes and its accusers. As in our scheme, an accusation is flooded in the network against a node detected misbehaving. If the number of accusers are at least k (the threshold) then the node is marked as convicted otherwise it is marked as a suspect. The certificate is not renewed for a convicted node. Hence, as opposed to our

model, there is no gradual build-up of trust in this model; instead, when a threshold k of accusation is received, the node is removed from the network.

Simulations show that as trust is gradually built up in the proposed model, fewer innocent nodes will have their certificate revoked and more malicious nodes will have their certificates revoked compared to the approach taken in [6]. Various attacks, Availability and Quality of service parameters are modeled in the simulation. The quality of service is defined by the response time. The performance of the proposed model and the model described in [6] under the influence of attacks is compared. The time taken to remove malicious nodes from the network by the two models is compared. The number of innocent nodes suffered in the two models is also compared. The amount of useful communication performed by the network in the two models is also compared.

4.1 Ad hoc network model

An ad hoc network consisting of a mixture of genuine and malicious nodes is modeled. Each node in the network has a unique ID. The nodes in the network are assumed to have entered the network and have acquired a certificate for communication. The Random waypoint mobility model [2] is chosen for this work. The threshold value 'k' is globally fixed. When two nodes communicate, the initiator of the communication will try to transfer a file to the destination. When a node wishes to communicate with a particular node in the network, it sends a request to the node and waits for the reply, where each node has its own reply time. A node may flood accusations or based upon its certificate validity it will communicate with other nodes in the network. If the certificate has expired it will send a request for renewal of the certificate. In the simulation of the Bayesian network trust model each node has its own weight for the two parameters of trust (availability, quality of service). For our simulations the two parameters are given equal weighting. The threshold values for satisfaction from a node is globally fixed.

The following additional assumptions are made for the simulation:

a) When the nodes initially enter into the network they are granted certificates for communication
b) Whenever a node receives a set of request for certificate, they are processed in the order they have arrived without giving any priorities.
c) When a node is under DoS attack it will be freed from the attack once it is out of the proximity of the attacker.
d) No nodes enter or leave the network for the duration of the simulation.
e) All nodes are assumed to have the same battery life and other resources.
f) When any node observes malicious activity, the node will flood the accusation in the network.
g) For simulation purposes, only one hop neighbors reply to certificate requests

The simulation program was written in the 'C' programming language on Microsoft.Net platform. A random waypoint model was used. For our simulations we used a constant file size of 100 KB, a network of 30 nodes in a 600m * 600m area, a pause time and a movement time with a maximum of 40 seconds, a data payload of 512 bytes per packet and a packet rate of 4 packets per sec. Each node has a communication range of 250m. The simulations were run for 800s. Each session lasts far 40 seconds. Portfolios are exchanged at the end of the session and the next session starts. In our model half the nodes are generating traffic. Portfolios, recommendations and accusation messages are assumed

to consume 1 sec each. A spoofing attack takes place through an intermediate malicious node in the path forwarding a message. A malicious node of each type (spoofing, DoS, hijacking) was randomly selected to generate an attack in each session. The attacks continued until the nodes were detected and removed from the network. A timer was invoked to measure the availability of a node. For quality of service, some nodes had a larger response time than other nodes. A recommendation was requested by one of the good nodes randomly. Accusations were flooded each time a malicious node was detected and portfolios were exchanged at the end of a session. The overheads simulated included the flooding of accusations, the exchange of portfolios and the reception of recommendations.

Two sets of simulations were run. In the first set 10% of the nodes were malicious, and in the second set 50% were deemed malicious. The attacker nodes were equally distributed between the different types of attacks. The simulations do not take into account routing overheads such as route discovery or packet losses. Future work will implement the proposed trust scheme on a network simulator such as ns-2.

4.2 Results

This section gives a comparison of the two trust models. In all the graphs below scheme1 refers to the distributed trust model described in [6] and scheme2 is the proposed trust model. We simulated a file transfer as the communication in the network.

4.2.1 Useful communication in the two trust models

Useful communication is defined as the communication that transfers files as opposed to communicating recommendations, portfolios, credentials or accusations. A simple 'back of the envelope' analysis for useful communications is shown below. Useful communication is calculated as follows:

Potential file transfer F_p*:* The file size that a node is capable of transferring based on its certificate validity period.

Potential file transfer rate F_p = *Download speed of a node* = S (packets/s)

However, due to overhead and other communications and the capacity of the receiver, the actual file size that is transferred may be less. We assume the a node can receive and transmit at the same rates and all nodes have the same capacity.

Actual File transfer: The file size that a node is able to send is calculated as:

Actual File transfer $F_a = F_p - O$ (16)

Where O is the overhead packets incurred.

Percentage of useful communication $= (F_a / F_p) * 100$ (17)

The actual file transfer is affected by the overheads

There are three overheads:

- Accusations. Given that on average the rate of accusations is λ accusations per second per node in the network and there are N nodes in the network where the average number of degrees of a node is δ, then the number of accusation packets generated per second in the network will be:

  $\lambda N \times N\delta = \lambda N^2\delta$ (18)

  Accusations are flooded in the network

- Recommendations. Given that on average the rate of recommendations is ε recommendations per second per node in the network, the number of recommendation packets generated per second in the network will be:

  εLN (19)

  where L is the average route length in the network. Recommendations are not flooded, and are sent directly to the requestor.
- Portfolios. Portfolios are exchanged at the end of a session. If N nodes are transferring a file of size M packets over a route of length L, the total packets transferred will be NLM. Taking into account the overheads, assume it takes γ time units to transfer the entire file. In other words,

  $$\gamma\lambda N^2\delta + \gamma\varepsilon LN + NLM = NS\gamma$$

  Therefore, per unit time:

  $$\lambda N^2\delta + \varepsilon LN + \frac{NLM}{\gamma} = NS \quad (20)$$

Assuming a rate of portfolio exchange σ, which is one every session, the above approximates to

$$\lambda N^2\delta + \varepsilon LN + \frac{NLM}{\gamma} + \frac{\sigma NL}{\gamma} \approx NS$$

Only $\frac{NLM}{\gamma}$ is useful file transfer, the rest is overhead. Therefore per node, the useful communications per unit time is $S - (\lambda N\delta + \varepsilon L + \frac{\sigma L}{\gamma})$ (21)

Therefore for each node the percentage of useful communication = (F_a / F_p) *100

$$= \frac{S - (\lambda N\delta + \varepsilon L + \frac{\sigma L}{\gamma})}{S} * 100 \quad (22)$$

This assumes that a file transfer takes more than one time unit.

However this does not take into account the overheads caused by mobility. Mobility causes link breakage and given a link breakage rate μ, the following occur

- Accusations. Accusations are re-flooded in the network at a rate that is related to λ and μ. The higher the value of μ, the higher the re-flooding rate
- Recommendations. Recommendation are re-transmitted in the network at a rate that is related to ε and μ. The higher the value of μ, the higher the re-transmission rate
- Portfolios. Portfolios are re-transmitted in the network at a rate that is related to σ and μ. The higher the value of μ, the higher the re-transmission rate

The mobility rate therefore affects the associated overheads. The higher the mobility, the higher the overheads. A detailed analysis is not given as our simulations did not cater for the extra overhead packets due to link breakage. In our simulations we have assumed that link breakage is handled at lower levels which will retransmit as needed.

From (22), we can determine that the longer the sessions, the smaller the overheads due to portfolio exchanges. The overheads due to accusations is related to the size of the network N. The rate of accusations λ is related to the number of malicious nodes in the network. The exact rate of accusations is a function of the trust mechanism employed. The

route length affects the overheads due to recommendations and portfolio exchanges. The frequency of attacks (as indicated by λ), recommendations requested (as indicated by ε), also affect the overheads.

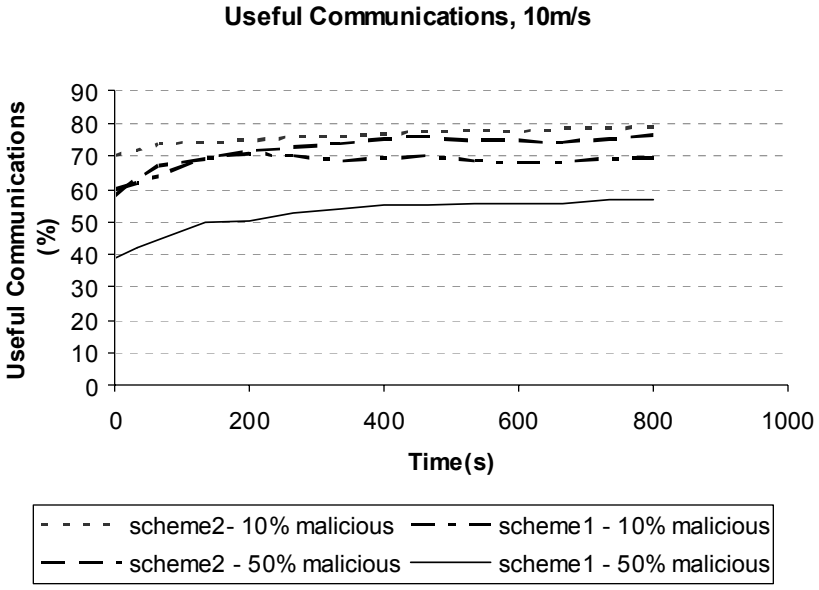

Figure 7: Comparison of useful communications at 10 m/s

Figure 7 compares the percentage of useful communication in the two models at 10m/s with different percentages of malicious nodes. The graph is as expected from the above simple analysis. From the graph above we can see that scheme2 achieves a higher percentage of useful communication when compared to scheme1 particularly when the percentage of malicious nodes in the network is high. This is because in scheme1 the nearest node is selected for communication whereas in scheme2 a node is selected based on its trust levels for availability and quality of service. The results also show that the overhead is higher when the number of malicious nodes is more. The speed of movement does not appear to impact the percentage of useful communications for scheme 2 as expected. This is because the lower level protocols compensate for packet losses caused by mobility. One would expect an increase in the overheads at high mobility if there is an exchange of portfolios every time a path breakage is expected. These results also indicate that although there is some overhead in the proposed approach, it is acceptable. The overhead is a function of the frequency of attacks and the number of malicious nodes in the network. The useful communication is lower initially because as expected, there is little prior experience in the network and more recommendations are requested. Moreover is scheme 2 malicious nodes are detected quickly resulting in many accusation messages initially. Once the network has settled down, the number of recommendations reduces. The results are very similar for 2m/s, but a little better. We have employed a high frequency of attacks and a large number of malicious nodes in the network. The overheads are therefore likely to be less in a more realistic scenario. The next set of results show that the proposed model performs well in the detection of malicious nodes

4.2.2 Number of true positives - Malicious nodes detected and removed

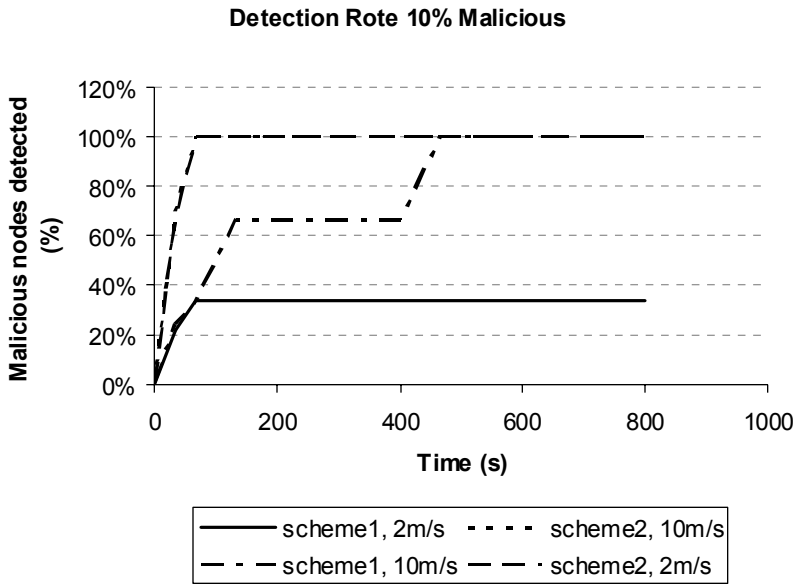

Figure 8: Comparison of malicious nodes detected at 10 % malicious nodes

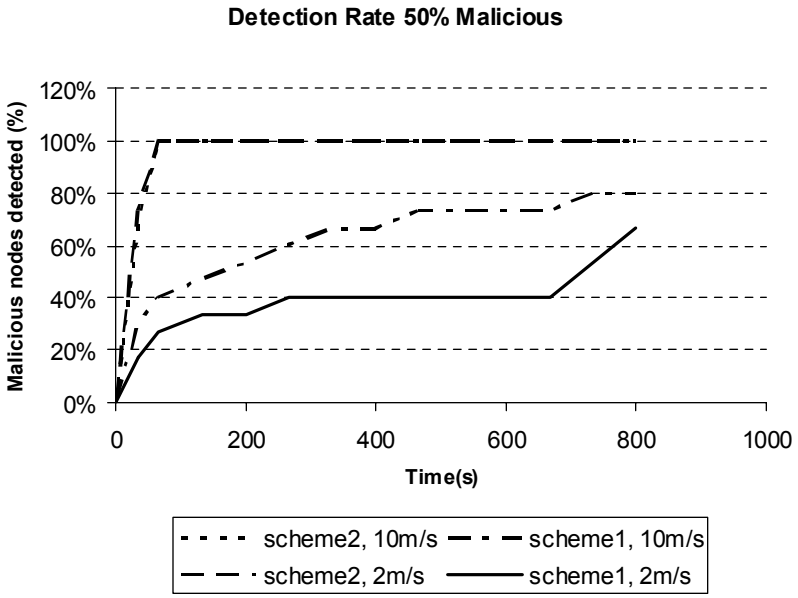

Figure 9: Comparison of malicious nodes detected at 50% malicious nodes

When 10% or 50% of the nodes are malicious, all the malicious nodes are removed in scheme2 – they are detected fairly rapidly and mobility of the nodes are does not seem to affect the rate of detection. Even at 50% all the malicious nodes are detected quickly. In scheme1 all the malicious nodes are nodes are not detected, particularly at low speeds. This is because at low speeds malicious nodes come into contact with a limited number of nodes

and the rate of detection is much slower. In scheme2, even if the mobility is low, a single trustworthy node may get enough experience to make an accusation and the malicious node will be detected, whereas in scheme1 at least k nodes have to make the accusation. Hence mobility is favorable to node detection in scheme 1.

4.2.3 Number of false positives – Good nodes detected and removed

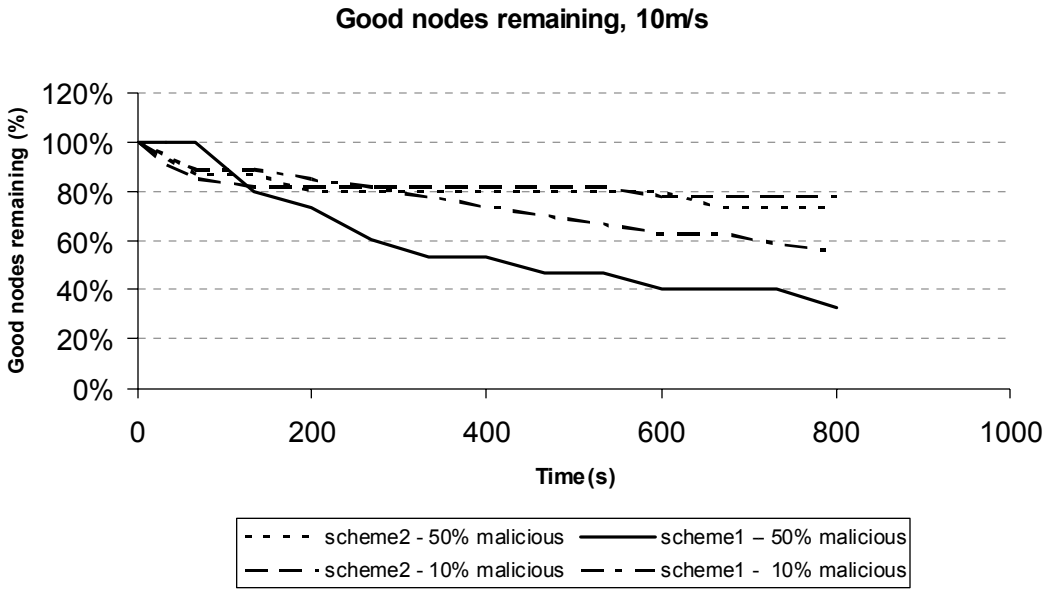

Figure 10: Comparison of good nodes detected as malicious at 10 m/s

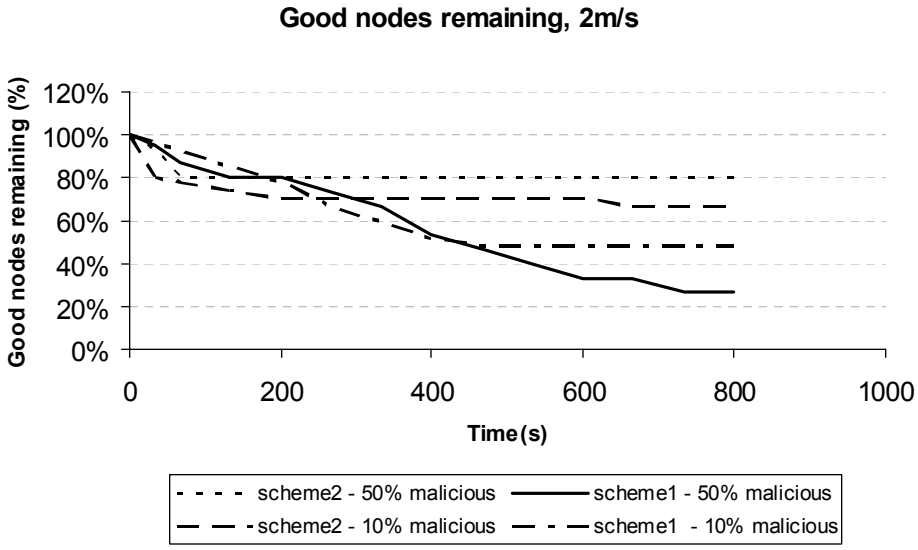

Figure 11: Comparison of good nodes detected as malicious at 2 m/s

The graph in figure 10 shows the number of well behaving nodes remaining in the two schemes at node mobility 10m/s. In both schemes a number of innocent nodes are wrongly classified as malicious and ejected from the network. The difference in the two schemes is striking with scheme2 having significantly fewer false positives than scheme1, particularly when the number of malicious nodes is high.

It is interesting to note that at low speeds (fig 11 - 2m/s), scheme1 performs better than scheme2 initially. This is because in scheme1 a node is not removed from the network unless k or more nodes accuse it. This takes time. In contrast in scheme2, a single message from a trusted node is sufficient to remove a node in the network. However as the network stabilizes, scheme2 shows fewer false positives. In contrast, at high speeds nodes come into contact with different nodes much more often. At high speeds, hijacking nodes have more opportunity to capture other nodes and flood the network with accusation messages This results in more nodes making accusations which means that at high speeds scheme1 performs poorly. The proposed scheme (scheme2) effectively removes hijacking nodes early in the communication. However, even in scheme2, the number of false positives is not completely eliminated and good nodes are detected as being malicious.

5 Conclusions

This work shows that the proposed network trust model provides a good trust model both from a service point of view (availability and quality) and a security perspective. Communicating with nodes that are within communications range and are trustworthy is preferred over communicating with nodes that are nearest neighbors, but less trustworthy. Certificate revocation is a function of quality of service and maliciousness. The percentage of useful communication in the network is improved by introducing context specific trust relationships among the nodes. The intrusion detection system of each node combined with the trust relationships with the other nodes effectively removes malicious nodes in the network. The objective of improving useful communications, maximizing the removal of malicious nodes and minimizing the removal of innocent nodes from the network is realized by the proposed trust model.

Future work will investigate adding more parameters to availability and quality of service for the derivation of trust. Not considered at all in this work is the expenditure of energy in this model. An optimization model for energy consumption and trust is needed. Performance overheads caused by the Bayesian model and the modeling of trust under the influence of noise is another area for future work. Fairness in the model needs to be considered. The extension of the model to cater for a network where credentials, accusations, and recommendations themselves cannot be trusted is an important area that needs more work. Other areas for future work include a more precise definition of experience and detecting nodes which may send only partial portfolios. Implementing the proposed scheme on a network simulator such as ns-2 or OPNET is also worth considering to take into account routing and other overheads including packet loss.

References

[1] Buchegger S, Le Boudec.J.Y, "The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks", *In the proceedings of WiOpt `03*, March 2003.

[2] Camp Tracy, Boleng Jeff and Davies Vanessa, "A survey of mobility models for ad hoc Network research", Dept. of Math. And Computer Sciences, Colorado School of Mines, Golden, CO., September 2002. http://toilers.mines.edu/papers/pdf/Models.pdf
[3] Capra Licia, "Engineering human trust in mobile system collaborations", *Proc. 12th International Symposium on the Foundations of Software Engineering,* pp 107-116. 2004,
[4] Cr´epeau, Claude and Davis Carlton R., "A certificate revocation scheme for wireless Ad hoc networks", *1st ACM workshop on Security of ad hoc and sensor networks*, pp 54-61, 2003.
[5] Goff T, Abu-Ghazaleh N B, Phatak D S and Kahvecioglu R, "Preemptive routing in ad hoc networks", *Proc. ACM MobiCom*, July 2001
[6] Luo Haiyun, Zerfos. P, Kong Jiejun, Lu Songwu, Zhang Lixia, "Self-Securing ad hoc wireless networks", *Proceedings IEEE Symposium on Computers and* Communications pp 567-574, July, 2002.
[7] Johnson David B, Maltz David A, and Broch Josh, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. in *Ad Hoc Networking,* edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.
[8] Navid Nikaein, Mobile Ad Hoc Networking & Computing at Eurecom, http://www.eurecom.fr/~nikaeinn/adhocNetworks/introduction.html,2001.
[9] Nekkanti Rajiv K and Lee Chung-wei, "Trust based adaptive on demand ad hoc routing protocol", *Proceedings of the 42nd ACM Southeast Regional Conference*, pp. 88 – 93,: 2004
[10] Pirzada Asad Amir and McDonald Chris," Establishing trust in pure ad hoc networks", *Proceedings of the 27th conference on Australasian computer science*, pp 47-54, 2004
[11] Ren Kui, Li Tieyan, Wan Zhiguo, Bao Feng, Deng Robert H and Kim Kwangjo, "Highly reliable trust establishment scheme in ad hoc networks", *Computer Networks*, 45, pp. 687–699, 2004
[12] Szabo Nick, Shamir's secret sharing, http://szabo.best.vwh.net/secret.html, 1997.
[13] Theodorakopoulos George and Baras John S,"Trust evaluation in ad hoc networks" *Proc ACM workshop on Wireless Security*, pp. 1-10, 2004
[14] Wang Yao, "Bayesian Network-Based Trust Model in Peer-to-Peer Networks", *Proceedings Workshop on Deception, Fraud and Trust in Agent Societies,* 2003
[15] WaveLAN/PCMIA Card User's Guide – Lucent Technologies
[16] Yan Zheng and Zhang Peng, "Trust Evaluation Based Security Solution in Ad Hoc Networks", *Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, 2003
[17] Zhaoyu Liu , Joy A W, Thompson R A, " A dynamic trust model for mobile ad hoc networks", *Proceedings.10th IEEE International Workshop on Future Trends of Distributed Computing Systems*, FTDCS 2004., pp. 80- 85, 2004.
[18] Zhu Feng, Mutka Matt and Ni Lionel, "Facilitating secure ad hoc service discovery in public environments", *Journal of Systems and Software*, 76, pp. 45–54, 2005
[19] Kagal Lalana, Finin Tim, and Joshi Anupam, "Trust-Based Security in Pervasive Computing Environments*, IEEE Computer*, Vol. 34, No. 12, pp. 154-157, 2001
[20] Giovanni Vigna, Sumit Gwalani, Kavitha Srinivasan, Elizabeth M. Belding-Royer, Richard A. Kemmerer, "An Intrusion Detection Tool for AODV-based Ad hoc Wireless Networks", *Proc 20th Annual Computer Security Applications Conference,* 2004
[21] J. Wright. Detecting Wireless LAN MAC Address Spoofing. White Paper, January 2003.
[22] Y. Zhang, W. Liu, W. Lou, and Y. Fang. "Location-based compromise-tolerant security mechanisms for wireless sensor networks" *IEEE Journal on Selected Areas in Communications*, Vol 24, No 2, 2006
[23] Theodore Pagtzis, Peter T. Kirstein, Stephen Hailes, Hossam Afifi, "Proactive seamless mobility management for future IP radio access networks", *Computer Communications*, Volume 26, Number 17, pp. 1975-1989, 2003